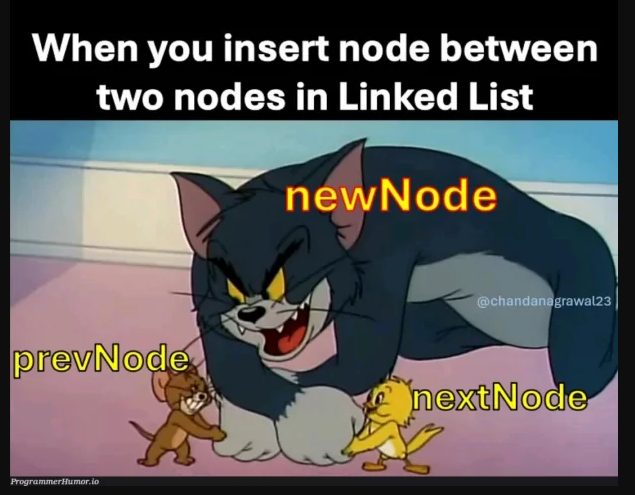
**MODUL 7**

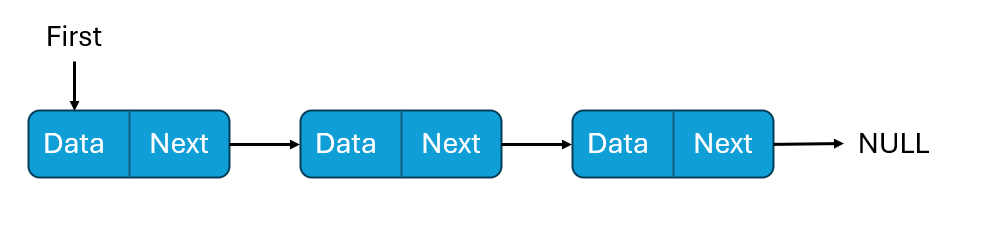
**LINKED LIST 2**

****

**Tujuan**

1. Praktikan dapat semakin memahami dan menguasai konsep linked list.
2. Praktikan dapat menggunakan pemahaman atas konsep linked list untuk mengembangkan linked list.

**Dasar Teori**

****

Modul Sebelumnya Praktikan telah diperkenalkan dengan konsep dari Linked List secara dasar dan Modul ini akan membahas Insert After dan Delete At.

Pada dasarnya, sebuah linked list beserta dengan data-datanya akan tersimpan secara dinamis dalam memory komputer karena menggunakan konsep malloc. Untuk mengakses data tersebut, linked list menggunakan pointer yang menunjuk ke tempat penyimpanan data tersebut. Sebuah elemen di dalam linked list terdiri dari data yang disimpan pada alamat elemen tersebut dan juga alamat untuk data pada elemen selanjutnya di dalam linked list.

Elemen dalam linked list akan berisi 2 tipe data, yaitu datanya sendiri sesuai dengan kebutuhan **(DATA)** dan alamat elemen selanjutnya pada linked list **(NEXT)**. Untuk memanggil elemen pertama, linked list akan menyimpan alamat dari elemen pertama dengan pointer pada bagian head. Untuk mengakses ke data selanjutnya, akan digunakan pemanggilan pointer Next yang menunjuk kepada elemen selanjutnya. Jika sudah berada pada elemen yang terakhir, pointer Next pada elemen tersebut akan menjunjuk ke NULL seperti yang ditunjukan pada gambar diatas.

**Operasi-Operasi pada Linked List:**

2(dua) Operasi yang paling penting pada linked list adalah insert dan delete dimana **Insert** ditujukan untuk memasukkan data pada linked list, dan **Delete** bertujuan untuk menghapus data dari linked list.

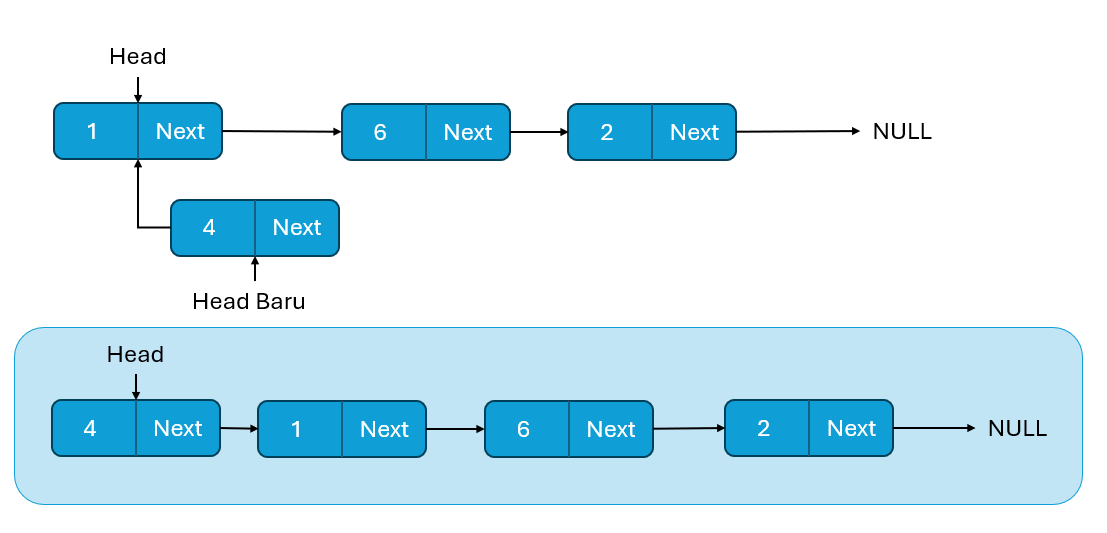
**INSERT**

Beberapa operasi insert pada linked list adalah sebagai berikut:

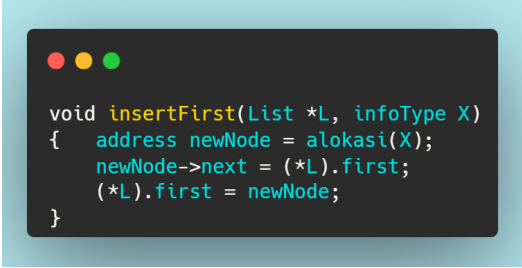
* InsertFirst
* InsertLast
* InsertAfter [New]

**InsertFirst** (memasukan data pada elemen utama)

A blue rectangle with black text

Description automatically generated

InsertFirst adalah operasi yang berfungsi untuk memasukkan elemen baru menjadi elemen pertama pada linked list. Jika linked list masih kosong, pointer pada head akan terhubung dengan alamat hasil alokasi elemen terbaru. Jika linked list sudah ada isi sebelumnya, maka pointer pada elemen baru akan terlebih dahulu dihubungkan ke data paling awal yang sebelumnya sudah ada, lalu pointer pada head akan dihubungkan dengan alamat hasil alokasi elemen terbaru.



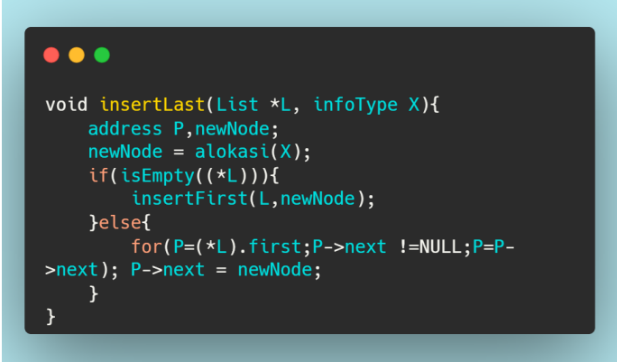
**InsertLast** (memasukan data sebagai elemen terakhir)

A blue rectangle with black text

Description automatically generatedA diagram of a computer

Description automatically generated

InserLast adalah operasi yang berfungsi untuk memasukkan elemen baru menjadi elemen terakhir pada linked list. Untuk memasukkan elemen menjadi elemen terakhir, perlu terlebih dahulu dicari data yang terakhir pada linked list. Setelah mencari elemen terakhir, pointer pada elemen terakhir tersebut akan terhubung dengan elemen yang terbaru. Jika belum ada elemen sama sekali di linked list, data di masukkan pada elemen pertama linked list seperti insertfirst.



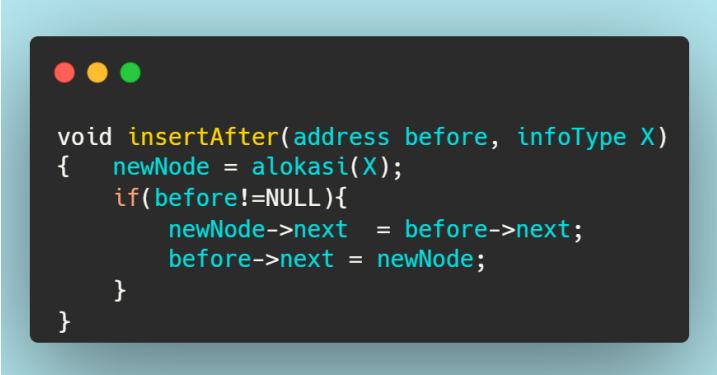
**InsertAfter** (Memasukkan data setelah data tertentu)

A blue rectangle with black text

Description automatically generatedA diagram of a computer program

Description automatically generated with medium confidence

InsertAfter adalah operasi untuk memasukkan elemen baru pada linked list setelah elemen tertentu. Untuk melakukan insertAfter, cari dulu elemen yang akan dimasukkan elemen baru setelahnya. Alokasikan elemen baru, dan hubungkan alamat next dari elemen sebelumnya ke alamat next dari temp, dan masukkan alamat next elemen sebelumnya menjadi alamat temp



**DELETE**

* DeleteFirst
* DeleteLast
* DeleteAt [New]

**DeleteFirst** (menghapus elemen pertama)

A blue rectangle with black text

Description automatically generatedA diagram of a flowchart

Description automatically generated

DeleteFirst adalah operasi yang digunakan untuk menghapus data pertama pada linkedList. Operasi ini akan melakukan penghapusan dengan terlebih dahulu menghubungkan linkedList pada elemen pertamanya kepada elemen setelah elemen pertama, atau elemen kedua. Lalu dilakukan pembebasan pada elemen pertama.

A computer screen shot of a program code

Description automatically generated

**DeleteLast** (Menghapus elemen terakhir)

A blue sign with black text

Description automatically generatedA diagram of a flowchart

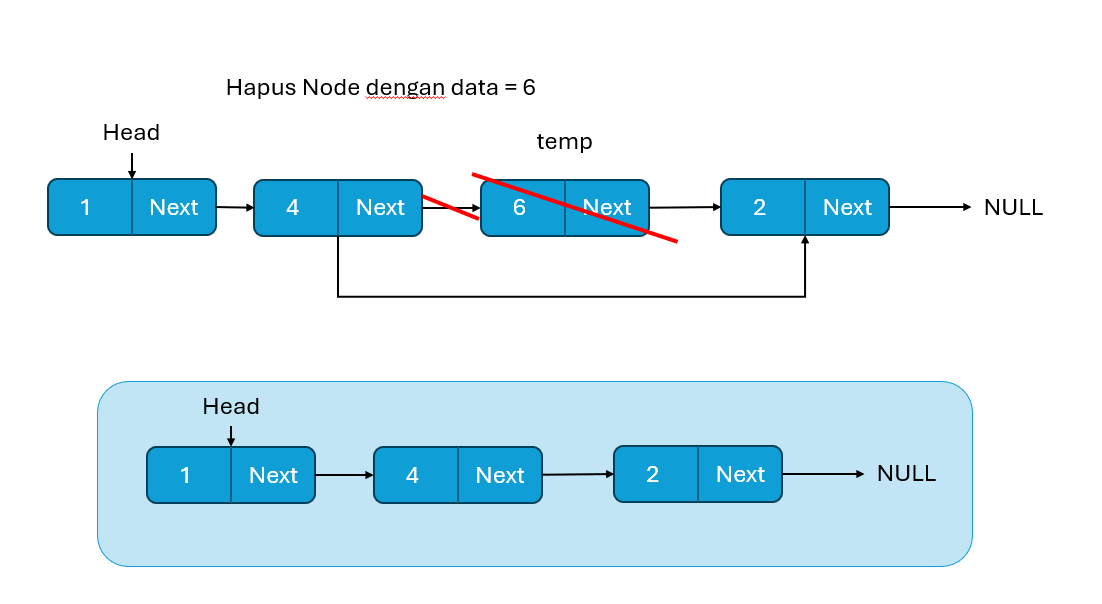
Description automatically generated

DeleteLast adalah operasi untuk menghapus data terakhir dari linked list. Cara menghapusnya adalah dengan mencari elemen sebelum elemen terakhir, lalu membebebaskan elemen terakhir. Elemen sebelum elemen terakhir tersebut akan dihubungkan ke null karena sudah tidak ada lagi data di elemen next/selanjutnya dari elemen tersebut

**A computer screen shot of a program code

Description automatically generated**

A blue background with black text

Description automatically generated**DeleteAt** (menghapus elemen tertentu)

Operasi DeleteAt akan menghapus data pada elemen tertentu. Untuk melakukan penghapusan ini, perlu dicari terlebih dahulu elemen sebelum elemen yang ingin dihapus. Elemen tersebut yang berada sebelum elemen yang ingin dihapus akan dihubungkan ke elemen selanjutnya dari elemen yang dihapus. Lalu hapus elemen yang ingin dihapus. Untu melakukan operasi ini, perlu digunakan operasi FindNode(Bisa dilihat di guided) untuk mencari data yang ingin dihapus terlebih dahulu.

**A screen shot of a computer program

Description automatically generated**

**GUIDED 7**

**Header.h**

**A screenshot of a computer program

Description automatically generated**

**Source.c**

**A screen shot of a computer program

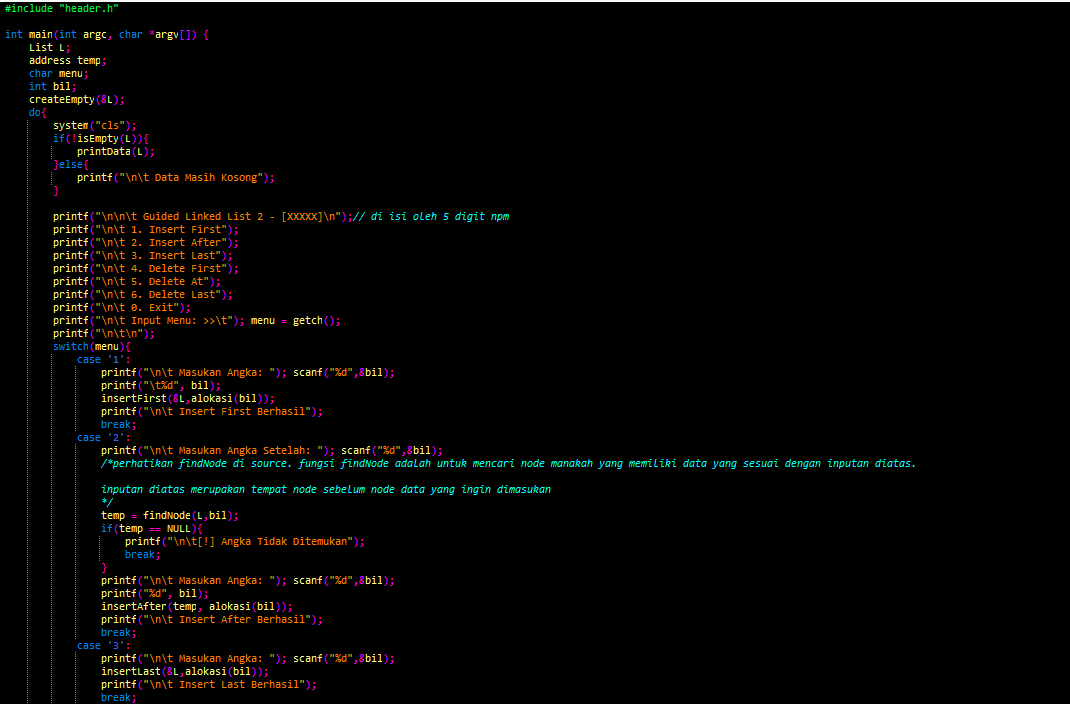
Description automatically generated**

**Source.c (Part 2)  
A computer screen with many white and green text

Description automatically generated with medium confidence  
Source.c (Part 3)A screen shot of a computer program

Description automatically generated**

**Main.c**

****

**Main.c (Part 2) A computer screen shot of text

Description automatically generated**

**KETENTUAN PENGUMPULAN**

Kumpulkan maksimal **H-1** SEBELUM PRAKTIKUM

Format pengumpulan:

**GD7\_X\_YYYYY**

X = kelas

Y = 5 digit NPM

Tips:

Pelajari cara memindah-mindahkan elemen dalam linked list